AUTOMATIC SPEAKER CLUSTERING

Hubert Jin, Francis Kubala, Rich Schwartz

BBN Systems and Technologies
Cambridge, MA 02138

ABSTRACT

This paper presents a fully automatic speaker clustering algorithm, which consists of three components: building a distance matrix based on Gaussian models of the acoustic segments; performing hierarchical clustering on the distance matrix with the prior assumption that consecutive segments should be more likely to come from the same speaker; and selecting the best clustering solution automatically by minimizing the within-cluster dispersion with some penalty against too many clusters. We applied this automatic speaker clustering technique in 1996 Hub4 evaluation, and the results show that it contributed significantly to the word error rate (WER) reduction in unsupervised adaptation. From our experiments, the algorithm seldom misclassifies segments from the same speaker into different clusters. We used the same clustering procedure for both partitioned evaluation (PE) and unpartitioned evaluation (UE) tests [1]. Experiments also show that this automatic speaker clustering algorithm improves unsupervised adaptation as much as the hand labeled ideal case where the clusters are generated based on true speaker, channel and background condition.

1. INTRODUCTION

During the last few years, it has been shown again and again that adaptation could significantly improve the performance of large vocabulary speech recognition systems. Today, almost all the advanced speech recognition systems come with speaker adaptation techniques. Speaker adaptation uses data from one speaker to move the parameters of the speaker independent system towards the speaker dependent values. Usually only a small amount of data is required for adaptation, although more data can always make the adaptation more robust. For unsupervised adaptation where the decoded transcriptions are used as the truth, we find out that adaptation with more data from the same speaker does help more in reducing word error rate.

Most of the previous speech recognition research deals with segmented speech such as the WSJ corpus, where the speaker and condition are constant over session. A cluster for any particular speaker can be generated by merging all the segments from the same speaker. However for many real-world continuous speech recognition problems, these are usually not available, nor are the boundaries of the speech. For continuous speech recognition such as the Hub4 evaluation, we need to segment half-hour audio programs and cluster the automatically segmented speech into speaker clusters. As an important component of the recognition systems, a good speaker clustering procedure can improve the performance of continuous speech recognition systems by supporting unsupervised adaptation. We have observed 10-25% relative WER reduction for unsupervised adaptation on a variety of tasks. Without the support of good clusters, the reduction could be smaller.

The goal of speaker clustering is to classify segmented speech into clusters such that each cluster contains speech from one speaker and also speech from the same speaker is classified into the same cluster. In practice, we regard speaker as a generic concept which really means speaker with channel and background condition. Thus, speech from the same physical speaker with significantly different channel and/or background conditions should be treated as speech from two different speakers in speaker clustering. On the other hand, we may want to classify speech from two speakers in the same cluster if their acoustic charactistics are not significantly different. In any case, the ultimate effectiveness of speaker clustering will be measured by how well the clusters do in adaptation.

We developed and implemented a speaker clustering algorithm which automatically determines all parameters based on a penalized model selection criterion. Our algorithm takes the advantage of the obvious fact that consecutive segments are more likely to come from the same speaker, but does not assume any prior knowledges about the speakers and their speech. We also introduce a penalty against too many clusters, so as to avoid the unwanted solution of one segment per each cluster. Both parameters, i.e. the measure of consecutive segments being from the same speakers and the number of total clusters, are data driven and the algorithm is fully automatic. Experiments show that this automatic speaker clustering algorithm improves unsupervised adaptation as much as the hand labeled ideal case where the clusters are generated based on true speaker, channel and background condition.

1996 Hub4 evaluation includes both the partitioned evaluation (PE) and the unpartitioned evaluation (UE) tests. In the PE test, the data was already partitioned into segments having constant speaker/channel/background conditions and each segment was given a feature label denoting these conditions. In the UE test, the speech needs to be segmented and the feature labels were not available. We believe the UE test is the real-world problem and chose to not to use the segmental feature labels in the PE test, in order to focus on approaches that would be viable for the general case. So the same speaker clustering procedure was used in both PE and UE Hub4 evaluation. In our Hub4 PE system, there is a procedure that chopped the original segments into shorter ones so that the BYBLOS decoder [4] could handle them more efficiently. From experiments on the development data, our speaker clustering algorithm seldom misclassifies chopped segments into different clusters due to ignoring the segmental labels in the PE test. This suggests that our speaker clustering procedure is reliable and should work well in the UE test too. In fact, our PE and UE systems are almost identical except the segmentation and gender detection procedures. The evaluation results show that the total degradation of our UE test from the PE test is only about 5% relatively.

In next section, we will describe the details of the speaker clustering algorithm. In section 3, some experimental results are provided to show the effectiveness of this algorithm. Finally in section 4, we will discuss other alternative model selection criteria, potential application of speaker clustering in speaker adapted training (SAT) [2] [3].

2. DESCRIPTION OF ALGORITHM

Consider that we have a collection of segments $S = \{s_1, s_2, ..., s_n\}$, and each s_i represents a sequence of spectral feature vectors, i.e. the Cepstral vectors in our implementation. Speaker clustering means to find a partition $P = \{p_1, p_2, ..., p_k\}$ of S such that each p_j contains only segments from the same speaker/condition and also speech segments from this speaker are classified into p_j only. Assume that the vectors in each of these sequences can be modeled as coming from a multivariate Gaussian distribution and that the vectors are statistically independent. A good clustering solution should have relatively small dispersion within clusters. The within-cluster dispersion [7] is defined as

$$W = \sum_{j=1}^{k} N_j * \Sigma_j$$

where Σ_j is the covariance matrix and N_j is the total number of feature vectors in cluster p_j.

There are several good clustering criteria [6]. We prefer to use the determinant of W to measure the goodness of speaker clustering. That is, the best clustering solution can be obtained by minimizing the measure over the parameter space. However in practice, this will usually lead to the unwanted clustering solution of one segment per cluster. Some penalty against too many clusters will help avoid the unwanted solutions. Thus, the best clustering solution will be obtained by minimizing the penalized measure instead.

There are three components in the implementation of the algorithm,

- building a **distance matrix** based on Gaussian models of the acoustic segments.
- performing **hierarchical clustering** to generate a list of clustering solutions.
- conducting **model selection** by the clustering criterion with a penalty against too many clusters.

2.1. Distance Matrix

Gish et al [5] introduced a distance measure between any two speech segments to reflect whether the two segments are from the same speaker. We use the same distance measure as the basis for the speaker clustering algorithm. However, noticing that consecutive segments are more likely to come from the same speaker, we prefer to scale the distance between consecutive segments by a parameter α whose value will be finally chosen in the model selection step.

2.2. Hierarchical Clustering

Cluster analysis [6] divides datapoints into clusters of points that are close to each other. The statistical software package *Splus* has these algorithms, such as *hclust*, implemented in its library. The hierarchical clustering procedure takes that distance matrix as input and continues to aggregate clusters together until there is just one big cluster. The output from the procedure is a tree of clusters. For any given number k, this cluster tree can be pruned subsequently with only k leaves left, which corresponds to the k tightest clusters in the solution. Here we treat each segment as an individual datapoint and the scaled distance measure as closeness between segments. Thus, for a list of combinations of (α, k), the cluster analysis procedure will generate a list of potential clustering solutions where each solution is a partition of $\{s_1, s_2, ..., s_n\}$. In the next step, the best solution will be chosen from this list based on the clustering criterion.

2.3. Model Selection

The idea of model selection is to find the best model from a list of potential models based on some criteria. Now we can regard each clustering solution as a model for the overall data. By the criterion of minimization of the determinant of the within-cluster dispersion matrix [6], a partition that minimizes

$$\mid W_{k,\alpha} \mid$$

is the best clustering solution. This criterion is one of the most favorite. But in practice, it is almost sure that it will end up with the unwanted solution of one segment per cluster, because the determinant measure will be non-increasing as the number of clusters increases. One approach to avoid this is to introduce a penalty against having too many clusters in the partition. So instead, we use the penalized criterion which will chose a partition $P_{k,\alpha}$ that minimizes

$$\mid W_{k,\alpha} \mid * \sqrt{k}$$

The figure 1 illustrates how the penalty helps to avoid the unwanted clustering solutions during model selection over the parameter space.

3. EXPERIMENTAL RESULTS

We did experiments to assess the effectiveness of the clustering algorithm on 1995 Hub4 development data. In that data set, segments are chopped into shorter ones, averaging about

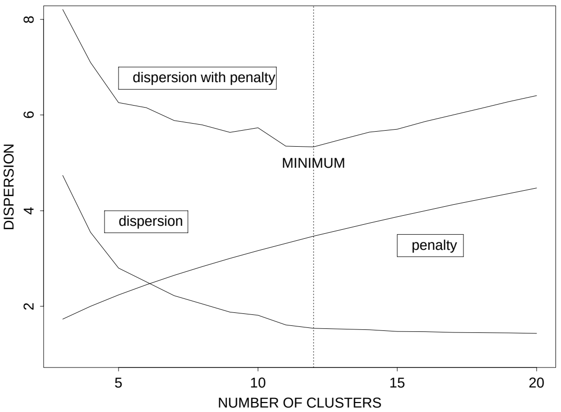

Figure 1: Model selection based on penalty against the number of clusters.

20 words per chopped segment. In PE segments adaptation, no clustering effort is needed and each cluster just includes only the chopped speech segments from the same original segment. However, this is still much better than the unwanted solution of one chopped segment per cluster, because the original segments in PE are usually long. The WER reduction of PE segments adaptation was 9% relatively from the baseline. As listed in table 1, clustering speakers by their true speaker, channel and background condition helped an extra 2% relative WER reduction. This motivated us to search for speaker clustering algorithms to reduce word error rate in the unsupervised adaptation step. With the support of the automatic speaker clustering, we observed a 12% relatively WER reduction from the baseline, which is even marginally better than that with the hand labeled ideal clusters. The automatic clustering was performed on the chopped segments. As expected, the scaling parameter α, which corresponding to the likelihood measure of consecutive chopped segments being from the same speaker, played an important role in our clustering algorithm. Without it in the algorithm, the WER reduction was not better than that of PE segments adaptation.

experiment	WER
baseline without adaptation	28.2
PE segments adaptation	25.6
clusters by true speaker-condition	25.0
chopped, clustered	25.8
chopped, clustered with adjacency	24.8

Table 1: Experimental results on one Marketplace episode, 940523, of Hub4 1995 (Phonetically Tied Mixture models).

The automatic clustering algorithm generated 25 clusters for the episode mentioned in table 1. Only 4 of them had segments with mixed speech conditions. Telephone/bandlimited speech was never mixed with anything else. This indicates that the blind clustering algorithm distinguish speech conditions quite well, especially for telephone/bandlimited speech.

For the PE development data, the segmentation was made manually with each segment having a feature label that is constant on speaker, channel and background. The hand segmentation practice has a tendency to make the segments as long as possible. This should help the WER reduction performance of PE segments adaptation in the PE, because each segment based cluster could actually contain several chopped segment from the same PE segment. However in the UE, the segmentation is done by the automatic segmentation procedure on the half-hour audio programs. So segment based adaptation actually means adaptation on clusters with only one chopped segment in each. Chopped segments are usually very short, i.e. less than 10 seconds or about 20 words. With less than 20 seconds of speech, unsupervised adaptation may not be robust enough. It is therefore expected that speaker clustering should help more, relative to segment based adaptation, in the UE than in the PE.

We applied this speaker clustering technique in 1996 Hub4 evaluation. Recently we did experiments to assess the performance of the automatic speaker clustering algorithm. The results, in table 2, shows that the automatic algorithm did only relative 0.7% worse than the hand labeled ideal case, where the relative WER was about 6.3% for the unsupervised adaptation.

experiment	WER
baseline without adaptation	31.83
clusters by true speaker-condition	29.73
chopped, clustered with adjacency	29.95

Table 2: Experimental results on all six development episodes of Hub4 1996 (State Clustered Tied Mixture models).

In our 1996 Hub4 PE system, there is a gender detection procedure prior to the speaker clustering. Only two segments were misclassified for their gender, but these wrong gender indicators were passed to the automatic speaker clustering procedure, which may explain some of the 0.7% degradation in relative WER reduction.

Although the automatic speaker clustering algorithm improves almost as much as the hand labeled ideal clustering based on speaker, channel and background condition, the model selection has a tendency to find less number of clusters than the truth as indicated in table 3.

episode	truth	algorithm
i960711p	13	6
k960712	30	21
l960710	62	30
n960715p	26	16
o960710p	15	9
p960712	22	17

Table 3: Number of clusters for each episode in 1996 Hub4 development data.

We found out that only seven original segments had their chopped segments not clustered together. The algorithm is in favor of putting speakers together and against splitting speech from the same speaker into different clusters. Since speech for some speakers in the episodes was very little, it might actually help reduce word error rate by merging speakers in the same cluster if their acoustic charactistics are not significantly different. It turns out that putting speakers together doesn't hurt.

4. DISCUSSION

We used the same speaker clustering procedure in both PE and UE tests. So in the PE test, speaker clustering procedure was done on the chopped segments and did not use any information from the original segments. The clustering algorithm improves as much as the hand labeled ideal true clustering based on speaker, channel and background condition. In general, this blind clustering algorithm distinguish speech conditions quite well, especially for telephone/bandlimited speech. However, as we mentioned in the previous section, it has a tendency to find a smaller number of clusters than the truth. In case where some speakers have very little speech, it might have been the right thing to let the speakers share the clusters. For the Hub4 systems, the ultimate effectiveness of speaker clustering is measured by how well the clusters do in unsupervised adaptation. From the experiments, our algorithm just works well. However for some applications, such as air traffic controller (ATC) and teleconference, where one of the goals is to label speech segments by speaker, to cluster speakers accurately is very important. It is worth investigating various combination of clustering criteria and penalties to search for potential better automatic speaker clustering algorithms. For example, some potential alternative criteria could be

$$\mid W_{k,\alpha} \mid + \mathrm{C} * \sqrt{\mathrm{k}}$$

or

$$\mid W_{k,\alpha} \mid + \mathrm{C} * \log \mathrm{k}$$

for some constant C.

Speaker clustering could also be used in speaker adapted training (SAT), similar to Padmanabhan's approach [8]. The training data of our Hub4 models includes speech from almost 2400 speakers, and most of them have less than 20 seconds total speech. Too little speech per speaker could cause unrobustness of the transformation matrix estimates in SAT training. Since speakers with the same condition labels can differ a lot acoustically, it might make sense to cluster these speakers by the automatic clustering algorithm and use the clusters as generic speakers in training.

5. CONCLUSIONS

We developed an automatic speaker clustering algorithm, and the same clustering procedure was used for both Hub4 PE and UE tests. Our experiments show that it improves WER as much as the hand labeled ideal clustering in unsupervised adaptation.

Acknowledgments

This work was supported by the Advanced Research Projects Agency and monitored by Ft. Huachuca under contract No. DABT63-94-C-0063. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

1. Kubala, F., H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, J. Makhoul, "The 1996 BBN BYBLOS Hub-4 Transcription System, *1997 DARPA Speech Recognition Workshop*, Chantilly VA, Feb. 1997, elsewhere this volume.
2. Anastasakos, T., J. McDonough, R. Schwartz, "A Compact Model for Speaker-Adaptive Training", *Proceedings of ICSLP-96*, Philadelphia PA, Oct. 1996.
3. Matsoukas, S., R. Schwartz, H. Jin, L. Nguyen, "Practical Implementations of Speaker-Adaptive Training", *1997 DARPA Speech Recognition Workshop*, Chantilly VA, Feb. 1997, elsewhere this volume.
4. Nguyen, L., R. Schwartz, "Efficient 2-Pass Nbest Decoder", *1997 DARPA Speech Recognition Workshop*, Chantilly VA, Feb. 1997, elsewhere this volume.
5. Gish, H., et al, "Segregation of Speakers for Speech Recognition and Speaker Identification," *IEEE International Conference on Acoustics, Speech, & Signal Processing Conference Proceedings,* 1991.
6. Everitt, B. *Cluster Analysis,* Halsted Press, New York, 1980, pp. 24-35.
7. Wilks, S., *Mathematical Statistics,* Wiley and Sons, New York, 1962.
8. Padmanabhan, M., et al, "Speaker Clustering and Transformation for Speaker Adaptation in Large-Vocabulary Speech Recognition Systems," *IEEE International Conference on Acoustics, Speech, & Signal Processing Conference Proceedings,* 1995, pp. 701-704.